

21 mars 2016



L3 Informatique 2015-2016

Université Paris 13

|  |  |
| --- | --- |
| **Elève** | Kevin GESNOUIN (gesnouin.kevin@gmail.com)  Alexis MALAMAS (malamas.alexis@gmail.com)  Loïc LAFONTAINE (lafontaine.loic@gmail.com)  Maxime LAVASTE (maxime.lavaste@gmail.com) |
| **Encadrant** | Nicolas ROLIN (nicolas.rolin@lipn.univ-paris13.fr)  Catherine RECANATI (cathy@lipn.univ-paris13.fr) |

Version 1.0f

Motus

Table des matières

[Introduction 2](#_Toc446960764)

[I. Présentation de Motus 3](#_Toc446960765)

[1. Règle du jeu Motus 3](#_Toc446960766)

[II. Description globale de Motus 4](#_Toc446960767)

[1. Profil des utilisateurs 4](#_Toc446960768)

[2. Environnement 4](#_Toc446960769)

[3. Hardware requis 4](#_Toc446960770)

[4. Outils requis 4](#_Toc446960771)

[5. Environnement de développement 5](#_Toc446960772)

[III. Les fonctionnalités 6](#_Toc446960773)

[1. Du côté utilisateur 6](#_Toc446960774)

[2. Du côté logiciel 6](#_Toc446960775)

[3. Interface graphique 6](#_Toc446960776)

[IV. Les modules 7](#_Toc446960777)

[1. Hiérarchie des modules 8](#_Toc446960778)

[2. Descriptions détaillées des modules 9](#_Toc446960779)

[Module « Paramètre » 9](#_Toc446960780)

[Module « Traitement d’un mot » 9](#_Toc446960781)

[Module « TALN » 10](#_Toc446960782)

[Module « Communication Client/serveur » 10](#_Toc446960783)

[Module « Interface Graphique et sonore » 10](#_Toc446960784)

[Module « Timer » 10](#_Toc446960785)

[Module « Jeu » 10](#_Toc446960786)

[V. Planification du projet 11](#_Toc446960787)

[VI. Améliorations possibles du projet 11](#_Toc446960788)

# Introduction

Dans le cadre du projet de fin d’études de la licence informatique proposé par l’Institut Galilée, notre but va être d’implémenter le jeu Motus de France 2 sur un support informatique pour qu’il soit jouable seul en mode entraînement. Motus est un jeu ludique dont le but est de trouver un mot mystère en un minimum de tentatives dans un temps imparti entre chaque proposition. Le joueur, dans sa réflexion, doit donc aussi considérer le temps qui lui reste.

Pour pouvoir développer Motus, nous devons dans un premier temps découvrir les particularités de Motus, dans un second temps, décrire les spécificités du logiciel, ainsi que les planifier et enfin choisir l’architecture et les langages informatiques que nous allons utiliser pour développer le jeu. De plus, nous avons convenu de la répartition des fonctionnalités du logiciel.

# Présentation de Motus

Motus est une adaptation française du jeu télévisé américain Lingo. Le programme est diffusé depuis le 25 juin 1990 sur Antenne 2 puis sur France 2. Le but du jeu est de découvrir des mots d’un nombre de lettres fixé.

Normalement, deux équipes de deux joueurs s’affrontent, mais nous allons uniquement développer un mode solo.

## Règle du jeu Motus

Le but du jeu est de trouver un mot mystère composé de 6 à 10 lettres. Les tentatives dont le joueur dispose sont définies en fonction de la longueur du mot.

Lors de chaque tentative, le joueur propose un mot appartenant à la langue française dans un délai imparti de 8 secondes. Les lettres bien placées sont alors indiquées par un carré rouge sur la grille du joueur, les lettres mal placées sont entourées d’un cercle jaune. La première lettre du mot à trouver est indiquée au joueur dès le début de la partie. La grille est de dimension taille du mot x nombre de tentatives.

La partie se termine soit lorsque le joueur a trouvé le mot, il est alors déclaré gagnant, soit lorsqu’il n’a pas trouvé le mot au bout de toutes ses tentatives.

# Description globale de Motus

La proposition de projet de départ demande une simulation du jeu Motus de France 2 pour pouvoir jouer sur une application ou une page web afin que les joueurs puissent s’amuser ou s’entraîner.

## Profil des utilisateurs

Motus est un jeu qui s’adresse à toute personne lettrée disposant d’un moyen de se connecter à Internet. Nous devons donc concevoir une application utilisable par tout le monde, sans pré requis de connaissances informatiques, pour que n’importe qui puisse jouer à Motus.

## Environnement

Motus devra être accessible sur un site web. Être accessible sur le web implique d’être disponible et utilisable sur tous types d’écrans, du smartphone au bureau. Nous allons donc devoir développer un site responsive design pour pouvoir répondre à cette problématique. C’est-à-dire que notre contenu va devoir s’adapter à la taille de l’écran pour pouvoir modifier l’affichage ainsi que l’expérience utilisateur pour pouvoir garantir un bon fonctionnement de l’application.

## Hardware requis

Pour jouer, l’utilisateur devra posséder, au minimum, un écran graphique pour qu’il puisse communiquer avec l’application, d’un clavier tactile, physique, visuel ou vocal pour que l’utilisateur interagisse avec l’application et d’une connexion internet pour pouvoir accéder au site web. De plus, une souris ou un Touchpad est vivement recommandé pour utiliser l’application sur un ordinateur, ainsi qu’une sortie audio pour les effets sonores.

## Outils requis

Le projet se réalisera sous la forme d’un site web, il sera accessible sur les navigateurs internet récents. Nous sommes actuellement en train de réfléchir si nous réservons un nom de domaine ainsi qu’un hébergement ou placer le site web sur le serveur de l’université. Un hébergement privé nous permettrait d’avoir plus de sûreté que l’hébergement de l’université qui fait souvent des maintenances. De plus, le réseau de l’université étant accessible que par le réseau interne, il est fort probable que le projet ne soit pas accessible depuis l’extérieur si nous choisissons cette solution.

Pour utiliser la fonctionnalité de traitement automatique, du langage naturel (TALN) afin de savoir si un mot est bien orthographié ne sera pas créé, mais récupéré sur un Framework déjà créé par un organisme extérieur.

# Les fonctionnalités

## Du côté utilisateur

* Choisir la longueur des mots à rechercher.
* Choisir le temps entre chaque tentative.
* Choisir le nombre de tentatives.
* Démarrer le jeu.
* Saisir un mot grâce à un clavier.
* Consulter les meilleurs scores.
* Enregistrer son score et son pseudonyme.
* Choisir une langue entre français et anglais.

## Interface graphique

* Afficher la grille.
* Afficher la première lettre du mot mystère.
* Afficher le temps restant pour chaque proposition.
* Afficher le nombre de tentatives restantes
* Afficher un input.
* Afficher victoire et défaite.
* Mise à jour de la grille en ajoutant le mot.
* Mise en surbrillance des lettres liées à un identificateur.

# Les modules

* Gestion partie

Jeu

* Indication du temps restant pour une proposition
* Afficher les propositions du joueur et les concordances avec le mot à chercher
* Afficher le nombre restant de propositions
* Construire et afficher la grille du jeu
* Afficher une victoire ou une défaite
* Présence d’un input contenant la réponse du joueur
* Effets sonores selon évènement

Interface Graphique et Sonore

* Gestion des meilleurs de score

Communication Client/serveur

* Vérification de l’existence du mot en français ou en anglais (selon la langue choisie)

Module TALN

* Modification de la taille du mot
* Modification nombre de tentatives
* Modification temps entre chaque tentative

Paramètre

* Vérifier regex.
* Les lettres bien placées sont marquées par un carré rouge.
* Les lettres existantes mais mal placées sont marquées par un rond jaune.

Traitement du mot

* Gérer le temps

Timer

* Editeur de texte
* Navigateur web
* Interpréteur script

Environnement de développement

## Hiérarchie des modules

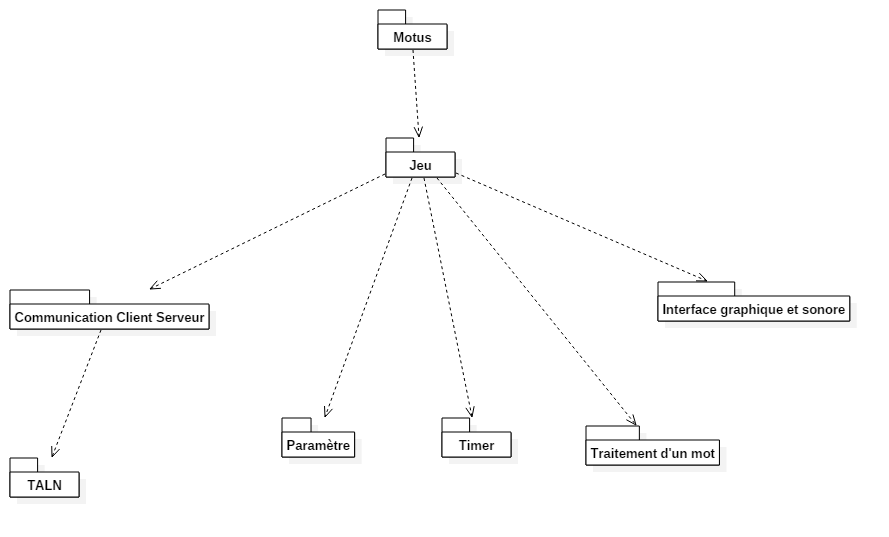


Figure 1 Hiérarchie des modules

## Descriptions détaillées des modules

### Module « Paramètre »

Le joueur a le choix de pouvoir régler la longueur des mots, le temps entre chaque tentative et le nombre de tentatives par mot. Lors de l’implémentation, nous avons le choix de l’utilisation de superglobales sur le serveur (utilisation des cookies et des sessions) ou conserver ces valeurs dans des variables côté navigateur.

### Module « Traitement d’un mot »

Il s’assurera de réagir à l’évènement « L’utilisateur entre un mot au clavier », puis mettra à jour l’interface graphique afin d’informer le joueur du résultat de sa tentative.

**Mot incorrect**

Figure 2 Processus de traitement d'un mot

Le mot entré devra respecter plusieurs contraintes pour passer à la vérification orthographique évaluée par une regex.

* Même longueur de mot que la taille demandée lors du réglage.
* Uniquement des caractères autorisés (26 lettres de l’alphabet).
* Le mot entré doit posséder la même première lettre que celle du mot mystère.

Concernant la vérification des correspondances avec le mot mystère, nous avons deux choix. Soit, lorsque l’on cherche un mot, on le conserve ensuite dans une variable chez le client, soit on le conserve côté serveur. Chaque méthode à ses défauts, la première étant qu’un joueur pourrait facilement tricher en regardant quel est le mot mystère dans la console. Pour la seconde, c’est que pour chaque demande de vérification, nous allons devoir utiliser une requête Ajax pour vérifier les concordances nécessitant des connexions récurrentes du joueur à notre serveur.

### Module « TALN »

Le TALN servira à vérifier l‘orthographe des mots. Ce module sera externe et nécessitera uniquement une implémentation dans notre projet. Nous allons effectuer des recherches pour savoir quel framework répond le plus à nos besoins. (Vérification orthographique française, libre de droits, rapide et peu lourd).

### Module « Communication Client/serveur »

À cela s'ajoute, la gestion des highscore. En effet, le module permet à un utilisateur de sauvegarder le nombre de mots consécutifs trouvés et de l’enregistrer. De plus, il peut consulter la liste des meilleurs scores.

Il utilisera des requêtes Ajax pour ne pas obliger les rechargements de pages afin de dynamiser notre site web.

### Module « Interface Graphique et sonore »

Le site web hébergeant le jeu sera mis en forme grâce au langage CSS. L'interface visible par l'utilisateur sera une grille de jeu, un bouton pour modifier la taille des mots, l'affichage d'un timer ainsi qu’un champ de formulaire afin de pouvoir rentrer un mot. Lorsque l’utilisateur rentre un mot, ce mot est affiché dans la grille de jeu avec le code couleur des bonnes lettres.

### Module « Timer »

Le Timer est la partie de l’application qui gère le temps. Il sera utilisé lors de la phase de réflexion entre chaque tentative du joueur pour savoir si le joueur dépasse la durée maximum pour proposer une proposition.

Pour l’implémenter, nous utiliserons les fonctions liées à la gestion de temps de JavaScript.

### Module « Jeu »

Le module jeu permet de rassembler tous les modules et de les faire communiquer entre eux. Donc, il gère le déroulement de la partie.

### Module « Environnement de développement »

Dans un premier temps, nous avons besoin d’un éditeur de texte ayant au minimum une coloration syntaxique telle que Notepad ou SublimeText pour développer notre application. Ensuite, un navigateur pour pouvoir tester l’affichage de notre site web ainsi que les scripts JavaScript. La console de développement web nous sert de débogueur.

Un interpréteur de script PHP côté serveur sera aussi nécessaire. Lors de la conception, nous utiliserons le logiciel Wamp, qui nous fournit un interpréteur PHP grâce aux modules d’Apache ainsi que MySQL pour la base de données.

Pour la création de ces modules, nous nous baserons sur les différents langages du web tel que HTML, PHP, JavaScript/Jquery.

# C:\Users\Kevin\Desktop\5767641a-786c-40b9-9432-4683285328ef.pngPlanification du projet

Figure  Diagramme de Gantt

# Améliorations possibles du projet

* Multijoueurs
* Mode ranking
* Multilingues
* Accessibilité pour les malvoyants